4

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

AD-A214 124

1b. RESTRICTIVE MARKINGS

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
1537622

5. MONITORING ORGANIZATION REPORT NUMBER(S)
N00014-86-K-0054

6a. NAME OF PERFORMING ORGANIZATION
University of Colorado

6b. OFFICE SYMBOL
(If applicable)

7a. NAME OF MONITORING ORGANIZATION
Office of Naval Research
Computer Science Division

6c. ADDRESS *(City, State, and ZIP Code)*
Campus Box 19
Boulder, CO 80309

7b. ADDRESS *(City, State, and ZIP Code)*
800 N. Quincy Street
Arlington, VA 22217-5000

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL
(If applicable)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
Young Investigator's Award and Distributed Self-Adaptive Databases

12. PERSONAL AUTHOR(S)
Roger King

13a. TYPE OF REPORT
King, Roger
FINAL

13b. TIME COVERED
FROM 9/30/85 TO 09/30/89

14. DATE OF REPORT *(Year, Month, Day)*
89/10/05

15. PAGE COUNT
12

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

The goal of the adaptive database project at the University of Colorado is to develop techniques which will make database systems useful for newer applications, such as engineering design. In such cases, there is a need to support complex, computed data, as well a need to hand-tailor a database system to suit specific processing requirements, for example, version support and document management. Two different experimental systems, one addressing each of these concerns, are under construction. The algorithms and techniques developed for these systems are intended to help relieve the advanced database user from the highly constrained mechanisms which traditional database management systems provide.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
□ UNCLASSIFIED/UNLIMITED □ SAME AS RPT. □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION

22a. NAME OF RESPONSIBLE INDIVIDUAL

22b. TELEPHONE *(Include Area Code)*

22c. OFFICE SYMBOL

Roger King
University of Colorado
303-492-7398
roger@boulder.colorado.edu
Young Investigator's Award and Distributed Self-Adaptive Databases
Contract N00014-86-K-0054, Final report (ending Sept. 88)
Contract N00014-88-K-0559, (period Oct. 88 - Sept. 89)

Numerical productivity measures

Papers in review (relating to the sponsored project)

1 for period of Oct. 88 to Sept. 89

Published papers (relating to the sponsored project)

4 for period of Oct. 88 to Sept. 89, 5 for Young Investigator award

Unrefereed articles

None.

Unpublished book portions

None.

Published book portions (relating to sponsored project)

4 for period Oct. 88 to Sept. 89

Patents filed

None.

Patents granted

None.

Invited presentations (relating to sponsored project)

2 for period Oct. 88 to Sept. 89, 13 for Young Investigator award

Contributed presentations

None.

Honors - during Young Investigator award

Best paper, Rocky Mountain Conference on Artificial Intelligence, June, 1987, "A Local Area Approach to Expert Database Systems" (with Jonathon Bein).

Prizes and awards

None.

Promotions

Promoted to Associate Professor, June 1988.

Graduate students supported

Young Investigator: 9 students for a total of 64 academic months (half time)

Oct. 88 - Sept. 89: 3 students for a total of 24 academic months (half time)

Post-docs supported

None.

Minorities supported

1 native born female phd student, currently working on thesis

Roger King
University of Colorado
303-492-7398
roger@boulder.colorado.edu
Young Investigator's Award and Distributed Self-Adaptive Databases
Contract N00014-86-K-0054, Final report (ending Sept. 88)
Contract N00014-88-K-0559, (period Oct. 88 - Sept. 89)

Summary of technical progress

The goal of the adaptive database project at the University of Colorado is to develop techniques which will make database systems useful for newer applications, such as engineering design. In such cases, there is a need to support complex, computed data, as well a need to hand-tailor a database system to suit specific processing requirements, for example, version support and document management. Two different experimental systems, one addressing each of these concerns, are under construction. The algorithms and techniques developed for these systems are intended to help relieve the advanced database user from the highly constrained mechanisms which traditional database management systems provide.

Roger King
University of Colorado
303-492-7398
roger@boulder.colorado.edu
Young Investigator's Award and Distributed Self-Adaptive Databases
Contract N00014-86-K-0054, Final report (ending Sept. 88)
Contract N00014-88-K-0559, (period Oct. 88 - Sept. 89)

Summary of technical results

1. Background

Traditionally, database systems were used by business programmers, and their needs were at least perceived to be rather simple. Data in the real world spans a wide spectrum of complexity, from highly unstructured (like text) to highly structured (like airplane designs). In data processing environments, data is typically represented only in a narrow band of this spectrum. All data is seen as being tightly, yet simply, structured. Further, most transactions against the database are submitted in batch mode. The goal is merely to support the fast retrieval of large numbers of similar, simply-structured records. As a result, conventional database systems provide very little in the way of abstraction, and in particular cannot effectively represent data whose internal structure is either highly structured or highly unstructured.

In recent years, a new generation of potential database users has emerged. This includes software engineers, VLSI and printed circuit board designers, aircraft and CAD engineers, as well as those involved in office automation. These individuals wish to store and manipulate many forms of data, in particularly, highly structured objects. (There is a need to represent unstructured data, specifically text, as well, but this research project does not address this issue.) Further, engineers often wish to manipulate data in an interactive environment. In sum, newer database users have a need for all the amenities a database system provides - such as concurrency, serializability, transaction management, rollback and recovery - but in an interactive design mode. Since traditional database systems do not suit these needs, many researchers are examining the numerous problems related to this grand challenge.

2. Research Objectives and Issues

Clearly, the goal of providing database support for interactive design users is gigantic. New data models, storage and access mechanisms, query languages, user interfaces, and many other tools are needed. In this project, we focus on two specific problems and use a common philosophical approach in attacking each of them. Our first area of concentration involves the support of computed data. In a design system, as opposed to a data processing system, there is a vast amount of tightly interconnected computed data. A design for an airplane includes highly interrelated data; changing one part of the design is likely to have effects on many other aspects. Further, it must be accessed quickly, as designers work in real time. Our second focus is on a broader issue, that of allowing advanced users to cleanly integrate into one database environment a variety of complex tools. For example, in a software development system used by software engineers, the database must interact with versioning, configuration, and report systems.

We are approaching each of these tasks from the perspective of adaptability. This means that, unlike existing database systems, the DBMS is not rigid. In our first research effort, we are focusing on the ability of the system to adapt itself at the physical level; computed data is managed in a way that allows the DBMS to learn from past usage experience and rearrange the way it processes updates. This is crucial in minimizing the potentially exponential costs of calculating computed data. In the second effort, we focus on the ability of the database user to adapt the system to suit his or her needs - at the conceptual level. This is important, as engineering applications vary dramatically in their requirements, and often require very specialized tools.

3. Approaches and Progress

The two projects described above are called Cactis and A La Carte. Cactis has resulted in the development of parallel algorithms for the maintenance of computed (or derived) data. These algorithms are based on attributed graphs and dramatically reduce the amount of I/O necessary to keep complex engineering database entities up to date. A La Carte uses the approach of abstracting the database management system up another level, resulting in the design of a database generator; such a system is, as a result, designed to be much more tailorable. The main problem lies in doing this in a fashion which does not require vast amounts of low-level programming.

Both of these projects also share another common philosophic approach, besides one of adaptability. They both attempt to integrate two directions which have been prominent in the database research community - behavioral and structural (or "semantic") object-oriented modeling. (Behavioral object-oriented modeling is often simply referred to by the term objected-oriented.) This has allowed the support of data objects which are both structurally complex and dynamic. This is crucial in supporting emerging engineering applications. Below, we discuss both projects, first Cactis, then A La Carte. We also briefly describe a system called FaceKit, which is a companion project to A La Carte, and is designed to provide user-adaptable graphical interfaces to databases.

3.1. Cactis

Consider an engineering design application familiar to all of us: software development and reuse. In every phase of the software life-cycle, we see a need for derived data. Examples include the following data relationships: the dependency between a source module and the corresponding object module; the derivation of a load module from a number of object modules; and, the relationships between a set of software modules and the associated documentation, requirements, bug reports, fix reports, and project milestones. In each case, if one piece of data changes, others are likely to be changed as a direct consequence.

With traditional database systems, this sort of derived data must be maintained by the application software or directly by end users - typically with a mechanism known as triggers. This introduces problems. Programmers are not likely write code that is portable from one software environment to another. Also, if computed data is maintained directly by the DBMS, then it may be managed in a much more efficient and correct fashion. Cactis [6, 8] is designed to support computed data in a highly efficient manner, and to do so in a consistent fashion. Triggers, on the other hand, must be hand-coded by the user and are difficult to reuse. Even more significantly, as a trigger mechanism is likely to operate in a first come, first severed basis, no attempt is made to optimize their execution. In general, if several trigger sequences all lead to the same piece of computed data, it could be updated an exponential amount of time, with respect to the number of trigger paths to the data item. A prototype Cactis system has been implemented, in order to provide a basis for the experimentation with and evolution of the underlying algorithms. In particular, substantial experiments have been performed in order to illustrate that the techniques developed are useful for engineering databases. The research is being conducted in conjunction with Scott Hudson of the University of Arizona.

Cactis represents a database as an attributed graph, and uses an incremental graph update algorithm. It also is self-adaptive, in that it learns from past experience and adjusts both process scheduling and data clustering on disk to minimize the I/O cost of maintaining computed data. We have run extensive performance tests on Cactis, illustrating substantial savings when the system is used. The potentially exponential behavior of triggers has been reduced to linear cost.

Also, several components of a software environment, including a "Make" [4] facility, a critical path tool, and a bug report system have been built on top of Cactis. Further, the Arcadia software environment project [5, 10, 11] has made some use of Cactis.

Cacti [7] is a distributed version of Cactis, and is currently under construction. It is targeted for a local network of Sun workstations, and is motivated by the fact that software design teams often work in distributed, interactive environments. The implementation of the system is being greatly facilitated by the fact that the graph algorithm in Cactis is naturally parallel, thus making it easy to adapt it to a distributed environment. In keeping with the self-adaptive nature of Cactis, the new system uses usage statistics to replicate, migrate, and recluster data around the network.

3.2. A La Carte

A La Carte [2] is in its early stages, and addresses much higher-level issues than Cactis or Cacti. The project, which is being conducted in conjunction with Colorado PhD students Pam Drew and Jonathan Bein, was motivated by the lesson that Cactis is still a very low level tool, and that many problems arise when trying to integrate various software environment tools within a Cactis application. Again, a prototype is under development, so that real experiments can be performed to validate and evolve the techniques under design.

The system uses mixins and multiple inheritance to allow an engineer to select both database facilities and software environment capabilities. For example, the designer of a software environment may choose an appropriate concurrency control option and clustering mechanism, as well as a version facility, a document management mechanism, and a configuration tool. A La Carte puts them all together in one system, using a method integration technique. It thus is very similar in spirit to Exodus [3] and Genesis [1]; a significant difference is that A La Carte is a less aggressive project, and is oriented mostly toward examining the appropriate mechanisms for resolving conflicts when mixing in complex software methods.

Users who wish to manipulate a database in a design environment are likely to desire a graphical interface. This, combined with the widespread availability of bit-mapped displays, has caused renewed interest in graphical interface design. There has been a lot of research recently in the construction of user interface management systems, but they are designed to handle general-purpose applications and are not tailored toward engineering database systems. In response to this, the FaceKit [9] interface design toolkit is under development. The goal is to build a user interface management system which, because it possesses specialized knowledge about databases (e.g., the notions of schema and query language) is able to more closely suit the needs of database users. The system is built on top of Cactis, and like A La Carte, gives the user a mechanism for adapting an interface to suit specific database needs.

References

1. D. S. Batory, J. R. Barnett, F. F. Garza, K. P. Smith, K. Tsukauda, B. C. Twichell and T. E. Wise, ''GENESIS: A Reconfigurable Database Management System'', *To appear in IEEE Transactions on Software Engineering*, .

2. J. Bein, P. Drew and R. King, ''A La Carte: A Generator of Persistent Object Stores for Software Environments'', *Working paper*, 1988.

3. M. J. Carey, D. J. DeWitt, D. Frank, G. Graefe, M. Muralikrishna, J. E. Richardson and E. J. Shekita, ''The Architecture of the EXODUS Extensible DBMS'', *Proceedings of the Workshop on Object-Oriented Databases*, Pacific Grove, California, September 23-26, 1986, 52-65.

4. S. I. Feldman, ''Make -- A Program for Maintaining Computer Programs'', *Software - Practice and Experience 9* (April 1979), 255-265.

5. D. Heimbigner and S. M. Sutton, ''SCAD Support for Software-Process Programming'', *ACM Software CAD Databases Workshop*, Napa, California, February 27-28, 1989.

6. S. Hudson and R. King, ''The Cactis Project: Database Support for Software Engineering'', *IEEE Trans. on Software Engineering*, June 1988.

7. S. Hudson and R. King, ''An Adaptive Derived Data Manager for Distributed Databases'', *Second International Workshop on Object-Oriented Databases*, Bad Munster am Stein-Ebernburg, FRG, September 1988, 193-203.

8. S. Hudson and R. King, ''Cactis: A Self-Adaptive, Concurrent Implementation of an Object-Oriented Database Management System'', *ACM Transactions on Database Systems*, 1989.

9. R. King and M. Novak, ''FaceKit: A Database Interface Design Toolkit'', *International Conference on Very Large Databases*, September, 1989.

10. S. M. Sutton, D. Heimbigner and L. J. Osterweil, ''Programmable Relations for Managing Change During Software Development'', *University of Colorado at Boulder Technical Report CU-CS-418-88*, September 15, 1988.

11. R. N. Taylor, F. C. Belz, L. Clarke, L. J. Osterweil, R. W. Selby, J. C. Wileden, A. Wolf and M. Young, ''The Arcadia Environment Architecture'', Tech. Report, Univ. of Calif.at Irvine, Dept.Info.and Comp. Sci, Irvine, Calif., Sept. 1987.

Roger King
University of Colorado
303-492-7398
roger@boulder.colorado.edu
Young Investigator's Award and Distributed Self-Adaptive Databases
Contract N00014-86-K-0054, Final report (ending Sept. 88)
Contract N00014-88-K-0559, (period Oct. 88 - Sept. 89)

Publications, presentations, and reports relating to project, for both contracts

Papers in review

"Performance of the Self-Adaptive Mechanisms for Maintaining Derived Data in Cactis", with Scott Hudson submitted to IEEE Transactions on Data and Knowledge Engineering

Published papers

"Cactis: A Self-Adaptive, Concurrent Implementation of an Object-Oriented Database Management System", with Scott Hudson, *ACM Transactions on Database Systems, December 1989.*

"Database Support for Software Environments", with Scott Hudson, *IEEE Transactions of Software Engineering,* June 1988.

"Semantic Modeling: Survey, Applications, and Research Issues", with Richard Hull, *ACM Computing Surveys,* September 1987.

"Three Software Environment Database Research Projects", *SCAD Support for Software-Process Programming,* Feb. 1989.

"The FaceKit Database Interface Generation Kit" with Mike Novak, *International Conference on Very Large Databases, August 1989.*

"An Adaptive Derived Data Manager for Distributed Databases", with Scott Hudson *Second International Workshop on Object-Oriented Databases,* Sept. 27-30, 1988.

"Freeform: A User-Adaptable Form Management System", with Michael Novak, *Proceedings of the Conference on Very Large Database Systems,* Brighton, England, September 1987.

"Object-Oriented Database Support for Software Environments", with Scott Hudson, *Proceedings of the ACM Sigmod International Conference on*

Management of Data, San Fiancisco, California, May 1987.

"Cactis: A Database System which Supports Functionally-Defined Data" with Scott Hudson, *proceedings of First International Workshop on Object-Oriented Database Systems,* September 1986.

Published book portions

"Object-Oriented Database Modeling and Software Environments", *Ada Reuse and Metrics,* edited by P.A. Lesslie, R.O. Chester, and M.F. Theofanos, 1989.

"Object-Oriented Database Tools to Support Software Engineering", with J. Bein and Pam Drew, *Ada Reuse and Metrics,* edited by P.A. Lesslie, R.O. Chester, and M.F. Theofanos, 1989.

"My Cat is Object-Oriented", *Object-Oriented Languages, Applications, and Databases,* W. Kim and F. Lochovsky, editors, Addison-Wesley, 1989.

"The Efficient Maintenance of Derived Data in Cactis", with Scott Hudson, to appear in *Object-Oriented Database Systems,* edited by Klaus Dittrich and Umeshwar Dayal, Springer-Verlag, 1990.

Invited presentations

"Database Support for Derived Data",
University of California at Irvine, Feb. 24, 1989.

"Distributed Database Support for Software Engineering,"
British Computer Society,
London, England, October 3, 1988.

"An Adaptive Derived Data Manager for Listributed Software Engineering Databases",
Ada Reuse and Metrics Workshop, U.S. Army Institute for Research in Management Information, Communications, and Computer Science,
Atlanta, Georgia, June 15, 1988.

"Database Support for Engineering Design",
DARPA - Future Database Directions Workshop,
Orange Grove, California, March 31, 1988.

"Database Support for the Software Life-Cycle",
Army Office for Information Management and Computer Science,
Atlanta, March 10, 1988.

"Object-Oriented Database Support for Software Environments",
Rocky Mountain AI Association,
Denver, July 7, 1987.

"Graphical Interfaces to Databases",
Advanced Data and Knowledge Base Systems,
Capri, Italy, June 11, 1987.

"A Self-Adaptive, Concurrent Object-Oriented DBMS",
Office of Naval Research,
Arlington Virginia, February 19, 1987.

"Semantic and Object-Oriented Database Systems",
John Hopkins Applied Physics Lab,
February 18, 1987.

"A Self-Adaptive, Concurrent Object-Oriented DBMS",
George Mason University,
Fairfax, Virginia, February 18, 1987.

"Object-Oriented Database Support for Software Environments"
Arcadia Software Environments Research Consortium
Laguna Beach, California, January 27, 1987.

"Cactis: A Database System for Specifying Functionally-Defined Data"
University of Florida,
Gainesville, Florida, November 19, 1986.

"SKI: The Semantically-Knowledgeable Interface"
IBM University Study Conference,
Fort Lauderdale, Florida, November 17, 1986.

University of Minnesota,
Computer Science Department, November 3, 1986.

"Distributed Object-Oriented Database Systems"
Naval Ocean Systems Research Center,
San Diego, California, September 8, 1986.

Roger King
University of Colorado
303-492-7398
roger@boulder.colorado.edu
Young Investigator's Award and Distributed Self-Adaptive Databases
Contract N00014-86-K-0054, Final report (ending Sept. 88)
Contract N00014-88-K-0559, (period Oct. 88 - Sept. 89)

Research transitions and DoD interactions

Naval Oceans Systems Center, San Diego:

NOSC contributed matching funds to my Young Investigator award. I visited them and discussed my technical results with Navy personnel. I also discussed their research goals and how they relate to the work I have done.

Texas Instruments, Dallas:

The adaptive clustering mechanism of Cactis (one of the two major contributions of the research) is being implemented in the Zeitgeist database system being developed by TI in Dallas. (See Zeitgeist: Database Support for Object-Oriented Programming by Ford et al., Springer Verlag Lecture notes, number 334, edited by K.R. Dittrich, 1988.)

Defense Advanced Research Projects Agency:

I attended the Darpa Future Database Directions Workshop on March 31, 1988. Several prominent database researchers shared their research results with each other and DARPA personnel, and discussed future trends of database research and funding. I presented the work I have performed for this contract.

Army Office for Information Management and Computer Science, Atlanta:

I have participated heavily in an AIRMICS effort to develop standards for software reusability and have contributed to two AIRMICS publications (see book chapters edited by Lesslie et al.). My work on engineering and self-adaptive databases played heavily in this task.

University applications:

Three university research groups have obtained the Cactis code for experimentation and possible use as a development platform for further research. I have not yet heard any results.

Roger King
University of Colorado
303-492-7398
roger@boulder.colorado.edu
Young Investigator's Award and Distributed Self-Adaptive Databases
Contract N00014-86-K-0054, Final report (ending Sept. 88)
Contract N00014-88-K-0559, (period Oct. 88 - Sept. 89)

Software and hardware prototypes

The Cactis database system, as described in the technical results section, is an approximately 80,000 line system (in C) which is reasonably robust (for an academic system). It has been used to build several prototype engineering database applications and extensive experimental runs have been performed to validate its performance.